Jets
Jet algorithms and jet substructure

Matteo Cacciari
LPTHE Paris
Université Paris Diderot

Includes material from
Gavin Salam and Grégory Soyez
Jet algorithms
  ▪ How jets are made

Background
  ▪ How to “clean them up”

Jet substructure
  ▪ What’s inside them
Why jets

A jet is something that happens in high energy events: a collimated bunch of hadrons flying roughly in the same direction.

We could eyeball the collimated bunches, but it becomes impractical with millions of events.

The classification of particles into jets is best done using a clustering algorithm.
Why do jets happen?

Gluon emission

\[ \int \alpha_s \frac{dE}{E} \frac{d\theta}{\theta} \gg 1 \]

Non-perturbative physics

\[ \alpha_s \sim 1 \]
Where are jets used?

- ATLAS and CMS have each published **400+** papers since 2010
  - More than **half** of these papers make use of **jets**
  - **60%** of the **searches** papers makes use of **jets**

(Source: INSPIRE. Results may vary when employing different search keywords)
Why are jets so important?
One purpose of a ‘jet clustering’ algorithm is to reduce the complexity of the final state, simplifying many hadrons to simpler objects that one can hope to calculate.
Jets can serve two purposes

- They can be **observables**, that one can measure and calculate
- They can be **tools**, that one can employ to extract specific properties of the final state

Different clustering algorithms have different properties and characteristics that can make them more or less appropriate for each of these tasks
Jet clustering algorithm

A **jet algorithm** maps the momenta of the final state particles into the momenta of a certain number of jets:

\[
\{p_i\} \xrightarrow{\text{jet algorithm}} \{j_k\}
\]

- particles, 4-momenta, calorimeter towers, ...

**Most algorithms contain a resolution parameter,** \( R \), **which controls the extension of the jet**

**Algorithm + parameter(s) + recombination scheme = jet definition**
Jet definitions as projections

Projection to jets should be resilient to QCD effects

Projections are NOT unique: a jet is NOT EQUIVALENT to a parton
Reconstructing jets is an ambiguous task

Jets are what we see.

2 clear jets

How many jets do you see?
Do you really want to ask yourself this question for $10^9$ events?

3 jets?
Reconstructing jets is an ambiguous task

2 clear jets

3 jets?
or 4 jets?
Reconstructing jets must respect rules

Collinear Safe

\[ \alpha_s^n \times (-\infty) \]

Infinities cancel

Collinear Unsafe

\[ \alpha_s^n \times (+\infty) \]

Infinities do not cancel

Perturbative calculations of jet observables will only be possible with collinear (and infrared) safe jet definitions.
Two main classes of jet algorithms

- **Sequential recombination algorithms**
  (also called hierarchical agglomerative clustering algs.)
  Bottom-up approach: combine particles starting from closest ones
  **How?** Choose a distance measure, iterate recombination until few objects left, call them jets
  Works because of mapping closeness ⇔ QCD divergence
  Examples: Jade, $k_t$, Cambridge/Aachen, anti-$k_t$, ….

- **Cone algorithms**
  Top-down approach: find coarse regions of energy flow.
  **How?** Find stable cones (i.e. their axis coincides with sum of momenta of particles in it)
  Works because QCD only modifies energy flow on small scales
  Examples: JetClu, MidPoint, ATLAS cone, CMS cone, SISCone….

Can be programmed to be fairly fast, at the price of being complex and often IRC unsafe (except SISCone)
Cone-type jets were introduced first in QCD in the 1970s (Sterman-Weinberg ’77)

In the 1980s cone-type jets were adapted for use in hadron colliders (SppS, Tevatron... ➔ iterative cone algorithms

LEP was a golden era for jets: new algorithms and many relevant calculations during the 1990s

- Introduction of the ‘theory-friendly’ $k_t$ algorithm
  - sequential recombination type algorithm, IRC safe
  - it allows for all order resummation of jet rates
- Several accurate calculations in perturbative QCD of jet properties: rates, jet mass, thrust, ....
\[ \text{Distance:} \quad y_{ij} = \frac{2 \min(E_i^2, E_j^2)(1 - \cos \theta_{ij})}{Q^2} \]

In the collinear limit, the numerator reduces to the \textit{relative transverse momentum} (squared) of the two particles, hence the name of the algorithm 

- Find the minimum \( y_{\text{min}} \) of all \( y_{ij} \)
- If \( y_{\text{min}} \) is below some jet resolution threshold \( y_{\text{cut}} \), recombine i and j into a single new particle (‘pseudojet’), and repeat
- If no \( y_{\text{min}} < y_{\text{cut}} \) are left, all remaining particles are jets
**e^{+}e^{-} k_{t} (Durham) algorithm in action**

Characterise events in terms of number of jets (as a function of $y_{\text{cut}}$)

Note that the **same** event can be seen to have **different** number of jets according to the value of $y_{\text{cut}}$

Resummed calculations for distributions of $y_{\text{cut}}$ doable with the $k_{t}$ algorithm
**e^+e^- k_t (Durham) algorithm v. QCD**

$k_t$ is a sequential recombination type algorithm

One key feature of the $k_t$ algorithm is its relation to the structure of QCD divergences:

$$\frac{dP_{k \rightarrow ij}}{dE_i d\theta_{ij}} \sim \frac{\alpha_s}{\min(E_i, E_j) \theta_{ij}}$$

The $y_{ij}$ distance is the inverse of the emission probability

- The $k_t$ algorithm roughly inverts the QCD branching sequence (the pair which is recombined first is the one with the largest probability to have branched)

- The history of successive clusterings has physical meaning
The LHC environment differs from the LEP one (and even the Tevatron) under many respects
Jet challenges at the LHC

The LHC environment differs from the LEP one (and even the Tevatron) under many respects

- Number of final state particles much larger (order $10^3$)
Jet challenges at the LHC

The LHC environment differs from the LEP one (and even the Tevatron) under many respects

- Number of final state particles much larger (order $10^3$)
- Many higher order calculations (NLO, NNLO) available
Jet challenges at the LHC

The LHC environment differs from the LEP one (and even the Tevatron) under many respects

- Number of final state particles much larger (order $10^3$)
- Many higher order calculations (NLO, NNLO) available
- Presence of background (underlying event and pileup)
Jet challenges at the LHC

The LHC environment differs from the LEP one (and even the Tevatron) under many respects

- Number of final state particles much larger (order $10^3$)
- Many higher order calculations (NLO, NNLO) available
- Presence of background (underlying event and pileup)
- Jets often initiated by a large-momentum heavy particle
Jet challenges at the LHC

The LHC environment differs from the LEP one (and even the Tevatron) under many respects

- Number of final state particles much larger (order $10^3$) ➔ needs a fast algorithm

- Many higher order calculations (NLO, NNLO) available

- Presence of background (underlying event and pileup)

- Jets often initiated by a large-momentum heavy particle
Jet challenges at the LHC

The LHC environment differs from the LEP one (and even the Tevatron) under many respects

- Number of final state particles much larger (order $10^3$)  
  ➞ needs a fast algorithm

- Many higher order calculations (NLO, NNLO) available  
  ➞ needs an IRC-safe algorithm

- Presence of background (underlying event and pileup)

- Jets often initiated by a large-momentum heavy particle
Jet challenges at the LHC

The LHC environment differs from the LEP one (and even the Tevatron) under many respects

- Number of final state particles much larger (order $10^3$) ➔ needs a fast algorithm

- Many higher order calculations (NLO, NNLO) available ➔ needs an IRC-safe algorithm

- Presence of background (underlying event and pileup) ➔ needs small/known susceptibility and/or ability to subtract background

- Jets often initiated by a large-momentum heavy particle
Jet challenges at the LHC

The LHC environment differs from the LEP one (and even the Tevatron) under many respects

- Number of final state particles much larger (order $10^3$) ➔ needs a fast algorithm

- Many higher order calculations (NLO, NNLO) available ➔ needs an IRC-safe algorithm

- Presence of background (underlying event and pileup) ➔ needs small/known susceptibility and/or ability to subtract background

- Jets often initiated by a large-momentum heavy particle ➔ needs capability to distinguish boosted object jet from QCD jet
Two parameters, $R$ and $p_{t,\text{min}}$
(These are the two parameters in essentially every widely used hadron-collider jet algorithm)

$$d_{ij} = \min(p_{ti}^2, p_{tj}^2) \frac{\Delta R_{ij}^2}{R^2}, \quad \Delta R_{ij}^2 = (y_i - y_j)^2 + (\phi_i - \phi_j)^2$$

Sequential recombination algorithm

1. Find smallest of $d_{ij}, d_{iB}$
2. If $ij$, recombine them
3. If $iB$, call $i$ a jet and remove from list of particles
4. repeat from step 1 until no particles left

Only use jets with $p_t > p_{t,\text{min}}$

Inclusive $k_t$ algorithm
S.D. Ellis & Soper, 1993
Catani, Dokshitzer, Seymour & Webber, 1993
The $k_t$ algorithm and its siblings

$$d_{ij} = \min(p_{ti}^{2p}, p_{tj}^{2p}) \frac{\Delta y^2 + \Delta \phi^2}{R^2}$$

$$d_{iB} = p_{ti}^{2p}$$

$p = 1$ $k_t$ algorithm

The $k_t$ algorithm and its siblings

\[ d_{ij} = \min(p_{ti}^{2p}, p_{tj}^{2p}) \frac{\Delta y^2 + \Delta \phi^2}{R^2} \quad \quad d_{iB} = p_{ti}^{2p} \]

\[ p = 1 \quad k_t \text{ algorithm} \]


\[ p = 0 \quad \text{Cambridge/Aachen algorithm} \]

Y. Dokshitzer, G. Leder, S. Moretti and B. Webber, JHEP 08 (1997) 001
The $k_t$ algorithm and its siblings

$$d_{ij} = \min(p_{t_i}^{2p}, p_{t_j}^{2p}) \frac{\Delta y^2 + \Delta \phi^2}{R^2} \quad d_{iB} = p_{t_i}^{2p}$$

$p = 1$  $k_t$ algorithm

$p = 0$  Cambridge/Aachen algorithm
Y. Dokshitzer, G. Leder, S. Moretti and B. Webber, JHEP 08 (1997) 001

$p = -1$  anti-$k_t$ algorithm
MC, G. Salam and G. Soyez, arXiv:0802.1189

In anti-$k_t$ pairs with a **hard** particle will cluster first: if no other hard particles are close by, the algorithm will give **perfect cones**

Quite ironically, a sequential recombination algorithm is the ‘perfect’ cone algorithm
IRC safety of generalised-k$_t$ algorithms

\[ d_{ij} = \min(p_{t_i}^{2p}, p_{t_j}^{2p}) \frac{\Delta y^2 + \Delta \phi^2}{R^2} \quad d_{iB} = p_{t_i}^{2p} \]

\p > 0

New **soft** particle ($p_t \to 0$) means that $d \to 0 \Rightarrow$ clustered first, no effect on jets

New **collinear** particle ($\Delta y^2 + \Delta \phi^2 \to 0$) means that $d \to 0 \Rightarrow$ clustered first, no effect on jets

\p = 0

New **soft** particle ($p_t \to 0$) can be new jet of zero momentum $\Rightarrow$ no effect on hard jets

New **collinear** particle ($\Delta y^2 + \Delta \phi^2 \to 0$) means that $d \to 0 \Rightarrow$ clustered first, no effect on jets

\p < 0

New **soft** particle ($p_t \to 0$) means $d \to \infty \Rightarrow$ clustered last or new zero-jet, no effect on hard jets

New **collinear** particle ($\Delta y^2 + \Delta \phi^2 \to 0$) means that $d \to 0 \Rightarrow$ clustered first, no effect on jets
<table>
<thead>
<tr>
<th><strong>IRC safe algorithms</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>$k_t$</strong></td>
</tr>
<tr>
<td>$d_{ij} = \min(p_{t_i}^2, p_{t_j}^2) \Delta R_{ij}^2 / R^2$</td>
</tr>
<tr>
<td>hierarchical in rel $p_t$</td>
</tr>
<tr>
<td>Catani et al ‘91</td>
</tr>
<tr>
<td>Ellis, Soper ‘93</td>
</tr>
<tr>
<td>NlnN</td>
</tr>
<tr>
<td><strong>Cambridge/Aachen</strong></td>
</tr>
<tr>
<td>$d_{ij} = \Delta R_{ij}^2 / R^2$</td>
</tr>
<tr>
<td>hierarchical in angle</td>
</tr>
<tr>
<td>Dokshitzer et al ‘97</td>
</tr>
<tr>
<td>Wengler, Wobish ‘98</td>
</tr>
<tr>
<td>NlnN</td>
</tr>
<tr>
<td><strong>anti-$k_t$</strong></td>
</tr>
<tr>
<td>$d_{ij} = \min(p_{t_i}^{-2}, p_{t_j}^{-2}) \Delta R_{ij}^2 / R^2$</td>
</tr>
<tr>
<td>gives perfectly conical hard jets</td>
</tr>
<tr>
<td>MC, Salam, Soyez ’08</td>
</tr>
<tr>
<td>(Delsart, Loch)</td>
</tr>
<tr>
<td>$N^{3/2}$</td>
</tr>
<tr>
<td><strong>SISCones</strong></td>
</tr>
<tr>
<td>Seedless iterative cone with split-merge</td>
</tr>
<tr>
<td>gives ‘economical’ jets</td>
</tr>
<tr>
<td>Salam, Soyez ‘07</td>
</tr>
<tr>
<td>$N^{2\ln N}$</td>
</tr>
<tr>
<td>‘second-generation’ algorithms</td>
</tr>
<tr>
<td>All are available in FastJet, <a href="http://fastjet.fr">http://fastjet.fr</a></td>
</tr>
<tr>
<td>(As well as many IRC unsafe ones)</td>
</tr>
</tbody>
</table>
Many ‘things’ can be clustered into (or lost from) a jet other than what we want (typically, perturbative radiation from a parton).

Ideally we’d like to be able to correct for these effects.
Pileup can deposit several tens of GeV (or even hundreds, in a heavy ion collision) into a medium-sized jet
Hard jets and background

How are the hard jets modified by the background?

**Susceptibility**
(how much bkgd gets picked up)

**Resiliency**
(how much the original jet changes)

**Jet areas**

**Backreaction**
Anti-\(k_t\) jets maximise resiliency, and their regular shapes makes them easier to correct for detector-related effects.

Default choice of all LHC collaborations.
“How (much) a jet changes when immersed in a background”

Without background
Resiliency: backreaction

“How (much) a jet changes when immersed in a background”

Without background
Resiliency: backreaction

“How (much) a jet changes when immersed in a background”
Resiliency: backreaction

“How (much) a jet changes when immersed in a background”

Without background

With background

Backreaction loss

Backreaction gain
Anti-\(k_t\) jets are much more resilient to changes from background immersion

(NB. Backreaction is a minimal issue in pp background and at large \(p_T\). Can be much more important in Heavy Ion collisions)
Hard jets and background

Modifications of the hard jet

\[ \Delta p_t = \rho A \pm (\sigma \sqrt{A} + \sigma_{\rho} A + \rho \sqrt{\langle A^2 \rangle - \langle A \rangle^2}) + \Delta p_t^{BR} \]

- **Background momentum density** (per unit area)
- **Background**
  - ‘susceptibility’
- **Back-reaction**
  - ‘resiliency’
Jet areas, graphically represented by the coloured regions, represent the susceptibility of each jet to contamination from diffuse, soft radiation. Given an IRC-safe jet algorithm, jet areas can be calculated numerically for each jet, opening the way for a jet-by-jet, rather than average, correction for background contamination.
**Observable level**

- Determination of *susceptibility to contamination* of each specific observable needed
- Possibility to get unbiased subtraction by construction
- Basic example: transverse momentum

\[ p_t^{\text{sub}} = p_t^{\text{raw}} - \rho A \]  
(MC, Salam 0707.1378)

- Other examples:
  - Analytical calculations of susceptibility for selected jet shapes (Sapeta et al. 1009.1143, Alon et al. 1101.3002)
  - Moments of jet fragmentation functions (MC, Quiroga, Salam, Soyez, 1209.6086)
  - Generic (numerical) approach to susceptibility determination for any shape (Soyet et al, 1211.2811)

**Event ( = particle) level**
Background subtraction

Observable level

- Determination of susceptibility to contamination of each specific observable needed
- Possibility to get unbiased subtraction by construction
- Basic example: transverse momentum
  \[ p_{t_{\text{sub}}} = p_{t_{\text{raw}}} - \rho A \]  
  (MC, Salam 0707.1378)

Other examples:
- Analytical calculations of susceptibility for selected jet shapes
  (Sapeta et al. 1009.1143, Alon et al. 1101.3002)
- Moments of jet fragmentation functions
  (MC, Quiroga, Salam, Soyez, 1209.6086)
- Generic (numerical) approach to susceptibility determination for any shape
  (Soyet et al, 1211.2811)

Event (\(=\) particle) level

- The event is modified before calculating observables (jets, shapes, etc)
- Method not naturally unbiased, but can often be tuned
- Final dispersion potentially lower, as effective number of particles usually reduced

Examples:
- CMS Voronoi method (Lai, unpubl.)
- Cleansing (Krohn, Schwartz, Low, Wang, 1309.4777)
- Constituent Subtraction (Berta, Spousta, Miller, Leitner, 1403.3108)
- PUPPI (Bertolini, Harris, Low, Tran, unpubl.)
- SoftKiller (MC, Salam, Soyez, 1407.0408)
- ....
Background subtraction: jet-based

Correction of a jet transverse momentum

\[ p_{T}^{\text{hard jet, corrected}} = p_{T}^{\text{hard jet, raw}} - \rho \times \text{Area}_{\text{hard jet}} \]

If \( \rho \) is measured on an event-by-event basis, and each jet subtracted individually, this procedure will remove many fluctuations and generally improve the resolution of, say, a mass peak.

\[ \Delta p_t = \rho A \pm (\sigma \sqrt{A} + \sigma_{\rho} A + \rho \sqrt{\langle A^2 \rangle - \langle A \rangle^2}) + \Delta p_t^{BR} \]

Irreducible fluctuations: uncertainty of the subtraction

Needs two ingredients: \( \rho \) and \( A_{\text{jet}} \)
Numerical jet shape correction

A generic jet shape (a function of the momenta of all constituents of a jet) is modified by the addition of pileup.

Correct it by calculating numerically the derivatives that enter its Taylor expansion and subtracting (this generalises the jet area/median subtraction for transverse mom.)

\[ V_{\text{jet,sub}} = V_{\text{jet}} - \rho V_{\text{jet}}^{[1]} + \frac{1}{2} \rho^2 V_{\text{jet}}^{[2]} + \cdots \]
Soft Killer introduces a particle momentum cut such that the median momentum density ($\rho$) of the event is zero.
Half of the event is empty $\Rightarrow \rho = 0$ (because it’s the median)

NB. SK needs tuning of the size of the patches used to calculate $\rho$.
0.4 was found to be a good choice for $R=0.4$ jets
**Soft Killer performance**

- **60 average PU events**
  - $\sqrt{s}=14$ TeV, $\mu=60$
  - Pythia(4C), noUE
  - anti-$k_t$(R=0.4)
  - $p_t>50$ GeV

- **$p_t$ shift, $\Delta p_t$**

- **$\Delta p_t$**

- **$\sigma_{\Delta p_t}$**

- **Area-median, noUE**
  - no PU correction
  - area-median
  - SoftKiller(0.4)

- **Area-median, UE**
  - Pythia(4C), noUE
  - anti-$k_t$(R=0.4), $p_{t,\text{jet}}>50$ GeV

- **SoftKiller(a=0.4), noUE**
  - SoftKiller(a=0.4), UE

- **$\Delta p_t$**
  - 60 average PU events

- **$\sigma_{\Delta p_t}$**
  - $\sqrt{s}=14$ TeV, Pythia8(4C)
  - anti-$k_t$(R=0.4), $p_{t,\text{jet}}>50$ GeV

Matteo Cacciari - LPTHE

Hard Probes - Wuhan - September 2016

38
Many jet shapes:
- jet mass
- kt clustering scale
- jet width (= broadening, = girth)
- energy-energy correlation moment
- $\tau_{21}$ and $\tau_{32}$ N-subjettiness ratios

Biases under control
Dispersions smaller than with other methods
Not all jets are created equal

For instance, you may want to be able to tell

Decay of a heavy (boosted) object

from this

Light parton fragmentation

Or, more generally, you may want to be able to tell something about how the jet originated (e.g. quark/gluon discrimination, quenching, ....)
How to ‘look’ inside a jet?

- Use the clustering history of a ‘physical’ hierarchical clustering algorithm

- Define jet shape-variables sensitive to specific distribution of radiation inside the jet

- Literally ‘look’ at the distribution of radiation inside the jet (machine-learning techniques)

- .....
The substructure of a jet can be exploited to

- tag a particular structure inside the jet, i.e. a massive particle
  - First examples: Higgs (2-prong decay), top (3-prong decay)

- remove background contamination from the jet or its components, while keeping the bulk of the perturbative radiation (often generically denoted as grooming)
  - First examples: filtering, trimming, pruning

To understand how we need to recall how a sequential recombination algorithm works
Dendrogram

Distance between two objects is given by the height of the lowest internal node that they share.

Order of clustering here is 1,2,3,4

The clustering sequence is 4-5 (1), 2-3 (2), 23-45 (3), 1-2345 (4)
anti-kt
Identifying jet substructure: try out anti-$k_t$

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. $z$).
Identifying jet substructure: try out anti-$k_t$

- anti-$k_t$ algorithm
- $\text{dmin is } dij = 3.57137 \times 10^{-5}$

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. $z$).
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. $z$).
Identifying jet substructure: try out anti-$k_t$

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. $z$).
Identifying jet substructure: try out anti-$k_t$

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. $z$).
Identifying jet substructure: try out anti-$k_t$ algorithm

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. $z$).
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).
Identifying jet substructure: try out anti-\(k_t\)

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. \(z\)).

Anti-\(k_t\) gradually makes its way through the secondary blob → no clear identification of substructure associated with 2nd parton.
Identifying jet substructure: try out anti-\(k_t\)

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. \(z\)).

Anti-\(k_t\) gradually makes its way through the secondary blob \(\rightarrow\) no clear identification of substructure associated with 2nd parton.
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

Anti-$k_t$ gradually makes its way through the secondary blob $\rightarrow$ no clear identification of substructure associated with 2nd parton.
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. \(z\)).

\textit{Anti-}k_t \textit{gradually makes its way through the secondary blob \(\rightarrow\) no clear identification of substructure associated with 2nd parton.}
Identifying jet substructure: try out anti-$k_t$

How well can an algorithm identify the "blobs" of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. $z$).

Anti-$k_t$ gradually makes its way through the secondary blob $\rightarrow$ no clear identification of substructure associated with 2nd parton.
Identifying jet substructure: try out anti-$k_t$

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. $z$).

Anti-$k_t$ gradually makes its way through the secondary blob $\rightarrow$ no clear identification of substructure associated with 2nd parton.
Identifying jet substructure: try out anti-$k_t$

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. $z$).

Anti-$k_t$ gradually makes its way through the secondary blob $\rightarrow$ no clear identification of substructure associated with 2nd parton.
Identifying jet substructure: try out anti-$k_t$

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. $z$).

Anti-$k_t$ gradually makes its way through the secondary blob → no clear identification of substructure associated with 2nd parton.
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

Anti-$k_t$ gradually makes its way through the secondary blob $\rightarrow$ no clear identification of substructure associated with 2nd parton.
Identifying jet substructure: try out anti-$k_t$

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. $z$).

Anti-$k_t$ gradually makes its way through the secondary blob $\rightarrow$ no clear identification of substructure associated with 2nd parton.
Second try

$kt$
Identifying jet substructure: try out $k_t$

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. $z$).
Identifying jet substructure: try out $k_t$

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. $z$).
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. $z$).
Identifying jet substructure: try out $k_t$

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. $z$).
Identifying jet substructure: try out $k_t$

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. $z$).
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

$k_t$ clusters soft “junk” early on in the clustering
Identifying jet substructure: try out $k_t$

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. $z$).

$k_t$ clusters soft “junk” early on in the clustering
Identifying jet substructure: try out $k_t$

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. $z$).

$k_t$ clusters soft “junk” early on in the clustering
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

\( k_t \) clusters soft “junk” early on in the clustering.
Identifying jet substructure: try out $k_t$

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. $z$).

$k_t$ clusters soft “junk” early on in the clustering
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. $z$).

$k_t$ clusters soft “junk” early on in the clustering.
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. $z$).

$k_t$ clusters soft “junk” early on in the clustering
Identifying jet substructure: try out $k_t$

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

$k_t$ clusters soft “junk” early on in the clustering.
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. z).

$k_t$ clusters soft “junk” early on in the clustering.

*Its last step is to merge two hard pieces. Easily undone to identify underlying kinematics*
Identifying jet substructure: try out $k_t$

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. $z$).

$k_t$ clusters soft “junk” early on in the clustering

*Its last step is to merge two hard pieces. Easily undone to identify underlying kinematics*
Identifying jet substructure: try out $k_t$

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. $z$).

$k_t$ clusters soft “junk” early on in the clustering

*Its last step is to merge two hard pieces. Easily undone to identify underlying kinematics*
Identifying jet substructure: try out $k_t$

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

This is crucial for identifying the kinematic variables of the partons in the jet (e.g. $z$).

$k_t$ clusters soft “junk” early on in the clustering

*Its last step is to merge two hard pieces. Easily undone to identify underlying kinematics*

This meant it was the first algorithm to be used for jet substructure.

Seymour '93
Butterworth, Cox & Forshaw '02
Splittings and distances

Quasi-collinear splitting ($p_{tj} < p_{ti}$)

Invariant mass:

$$m^2 \simeq p_{ti} p_{tj} \Delta R_{ij}^2 = (1 - z) z p_t^2 \Delta R_{ij}^2$$

$k_t$ distance:

$$d_{ij}^{(p_{tj} < p_{ti})} = z^2 p_t^2 \Delta R_{ij}^2 \simeq \frac{z}{1 - z} m^2$$

For a given mass, the background (parton shower) will have smaller distance $d_{ij}$ than the signal (symmetric $1 \rightarrow 2$ decay), i.e. it will tend to cluster earlier in the $k_t$ algorithm.

Potential tagger: last clustering in $k_t$ algorithm

This is where the hierarchy of the $k_t$ algorithm becomes relevant. QCD radiation is clustered first, and only at the end the symmetric, large-angle splittings due to decays are reclustered.
Cambridge/Aachen
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?
Identifying jet substructure: Cam/Aachen

**Cambridge/Aachen algorithm**

$$\Delta R_{ij} = 0.214286$$

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

Cambridge/Aachen algorithm

$p_t/\text{GeV}$

$0$ $1$ $2$ $3$ $4$ $y$

$0$ $10$ $20$ $30$ $40$ $50$
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

Cambridge/Aachen algorithm

\[ \Delta \text{R}_{ij} = 0.415037 \]
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?
Identifying jet substructure: Cam/Aachen

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

Cambridge/Aachen algorithm

\[ \Delta R_{ij} = 0.686928 \]
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

C/A identifies two hard blobs with limited soft contamination.
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

C/A identifies two hard blobs with limited soft contamination, joins them.
Identifying jet substructure: Cam/Aachen

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

C/A identifies two hard blobs with limited soft contamination, joins them.
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

C/A identifies two hard blobs with limited soft contamination, joins them, and then adds in remaining soft junk.
Identifying jet substructure: Cam/Aachen

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

C/A identifies two hard blobs with limited soft contamination, joins them, and then adds in remaining soft junk.
How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

C/A identifies two hard blobs with limited soft contamination, joins them, and then adds in remaining soft junk.
Identifying jet substructure: Cam/Aachen

Cambridge/Aachen algorithm

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

C/A identifies two hard blobs with limited soft contamination, joins them, and then adds in remaining soft junk.
Identifying jet substructure: Cam/Aachen

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

C/A identifies two hard blobs with limited soft contamination, joins them, and then adds in remaining soft junk.
Identifying jet substructure: Cam/Aachen

How well can an algorithm identify the “blobs” of energy inside a jet that come from different partons?

C/A identifies two hard blobs with limited soft contamination, joins them, and then adds in remaining soft junk.

The interesting substructure is buried inside the clustering sequence — it’s less contaminated by soft junk, but needs to be pulled out with special techniques.

Butterworth, Davison, Rubin & GPS ’08
Kaplan, Schwartz, Reherman & Tweedie ’08
Butterworth, Ellis, Rubin & GPS ’09
Ellis, Vermilion & Walsh ’09
Hierarchical substructure

anti-$k_t$ algorithm

$k_t$ algorithm

Cambridge/Aachen

$p_t$/GeV

y
Hierarchical substructure

Undo the last clustering step(s)
### The IRC safe algorithms

<table>
<thead>
<tr>
<th></th>
<th>Speed</th>
<th>Regularity</th>
<th>UE contamination</th>
<th>Backreaction</th>
<th>Hierarchical substructure</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_t$</td>
<td>☺☺☺</td>
<td>☁</td>
<td>☂</td>
<td>☁</td>
<td>☺☺☺</td>
</tr>
<tr>
<td>Cambridge /Aachen</td>
<td>☺☺☺</td>
<td>☁</td>
<td>☂</td>
<td>☁</td>
<td>☺☺☺</td>
</tr>
<tr>
<td>anti-$k_t$</td>
<td>☺☺☺</td>
<td>☻/☻</td>
<td>☁</td>
<td>☀</td>
<td>☺</td>
</tr>
<tr>
<td>SISCone</td>
<td>☀</td>
<td>☁</td>
<td>☻/☻</td>
<td>☁</td>
<td>☒</td>
</tr>
</tbody>
</table>

Array of tools with different characteristics. Pick the right one for the job.
‘Jet substructure’ papers in INSPIRE

Number of papers containing the words ‘jet substructure’

More than 100 papers since 2008
(+ some background noise)

Pioneered by M. Seymour in the early ‘90s, rebooted by BDRS paper

Papers containing "jet substructure"
+ pioneering papers by Mike Seymour in 1991 and 1994
(Source: INSPIRE)

15. Jet substructure as a new Higgs search channel at the LHC.
Published in Phys.Rev.Lett. 100 (2008) 242001
A two-prong tagger/groomer for boosted Higgs, which
- Uses the **Cambridge/Aachen** algorithm (because it’s ‘physical’)
- Employs a **Mass-Drop** condition, as well as an **asymmetry cut** to find the **relevant splitting** (i.e. ‘tag’ the heavy particle)
- Includes a post-processing step, using ‘**filtering**’ (introduced in the same paper) to clean as much as possible the resulting jets of UE contamination (‘**grooming**’)

\[ pp \rightarrow ZH \rightarrow \nu \bar{\nu} b \bar{b} \]
\[ pp \rightarrow ZH \rightarrow \nu\bar{\nu}b\bar{b} \]

Start with the hardest jet

Use C/A with large \( R = 1.2 \)

\[ m_j = 150 \text{ GeV} \]
undo last step of clustering

Check how the mass splits between the two subjets ($m_1 = 139 \text{ GeV}, m_2 = 5 \text{ GeV}$) and how asymmetric the splitting is.

If

$$\frac{\max(m_1, m_2)}{m_j} > \mu \quad \text{or} \quad \frac{\min(p_{t1}^2, p_{t2}^2)}{m_j^2} \Delta R_{12}^2 < y_{cut}$$

repeat
$$pp \rightarrow ZH \rightarrow \nu \nu bb$$

Stop when a large mass drop is observed (and recombine these two jets)

$\mu = 0.67$ and $y_{cut} = 0.09$
Start with the recombined jet.

The process is described as:

\[ pp \rightarrow ZH \rightarrow \nu\nu bb \]
Recluster the constituents with $R_{\text{filt}}$
The low-momentum stuff surrounding the hard particles has been removed
Cluster with a large $R$

Undo the clustering into subjets, until a large asymmetry/mass drop is observed: tagging step

Re-cluster with smaller $R$, and keep only 3 hardest jets: grooming step

pp $\rightarrow$ ZH $\rightarrow$ $\nu\bar{\nu}bb$

Butterworth, Davison, Rubin, Salam, 2008
First taggers/groomers

- **Mass Drop + Filtering**
  - Butterworth, Davison, Rubin, Salam, 2008
  - Decluster with mass drop and asymmetry conditions
  - Recluster constituents into subjets at distance scale $R_{\text{filt}}$, retain $n_{\text{filt}}$ hardest subjets

- **Jet ‘trimming’**
  - Krohn, Thaler, Wang, 2009
  - Recluster constituents into subjets at distance scale $R_{\text{trim}}$, retain subjets with $p_{t,\text{subj}} > \varepsilon_{\text{trim}} p_{t,\text{jet}}$

- **Jet ‘pruning’**
  - S. Ellis, Vermilion, Walsh, 2009
  - While building up the jet, discard softer subjets when $\Delta R > R_{\text{prune}}$
    and $\min(p_{t1}, p_{t2}) < \varepsilon_{\text{prune}} (p_{t1} + p_{t2})$

  **Aim:** limit contamination from QCD background while retaining bulk of perturbative radiation

  Trimming and pruner are a priori groomers, but can become taggers when combined with an invariant mass window test

  (if you can groom away everything then there’s no heavy particle in the jet)
Soft Drop declustering

Larkoski, Marzani, Soyez, Thaler, 2014

Decluster and drop softer constituent unless

\[
\text{Soft Drop Condition: } \frac{\min(p_{T1}, p_{T2})}{p_{T1} + p_{T2}} > z_{\text{cut}} \left( \frac{\Delta R_{12}}{R_0} \right)^\beta
\]

i.e. remove wide-angle soft radiation from a jet

The paper contains

✓ analytical calculations and comparisons to Monte Carlos
✓ study of effect of non-perturbative corrections
✓ performance studies

Example of SoftDrop performance when used as a boosted W tagger
The jet substructure maze

Some of the tools developed for boosted W/Z/H/top reconstruction

Matrix–Element

Jet Declustering

- Seymour93
- YSplitter
- Mass–Drop+Filter
- ATLASTopTagger
- JHTopTagger
- CMSTopTagger
- HEPTopTagger (+ dipolarity)
- Trimming
- Planar Flow
- Pruning
- Twist
- Trimming
- CoM N–subjettiness (Kim)
- N–subjettiness (TvT)
- ACF
- N–jetiness
- Shower Deconstruction

Qjets

Multi–variate tagger

Slide by G. Salam, now a few years old
Alternatives to hierarchical substruct.

- If what we are interested in is the structure of the constituents of a jet, the "jet" itself is not the most important feature.
- A different algorithm, or simply the study of the constituents in a certain patch will also do. Selected alternatives are:
  - Use of jet-shapes to characterise certain features
    - e.g. \textit{N-subjettiness}: how many subjets a jets appears to have
    - Thaler, van Tilburg, 2011
  - Alternative ways of clustering
    - e.g. \textit{Qjets}: the clustering history not deterministic, but controlled by random probabilities of merging. Can be combined with, e.g. pruning
    - Ellis, Hornig, Roy, Krohn, Schwartz, 2012
  - Use information from matrix element
    - e.g. \textit{shower deconstruction}: use analytic shower calculations to estimate probability that a certain configuration comes from signal or from background
    - Soper, Spannowsky, 2011
  - Use event shapes mimicking jet properties
    - e.g. \textit{JetsWithoutJets}, mimicking trimming
    - Bertolini, Chen, Thaler, 2013
\[ \tau^{(\beta)}_N = \sum_i p_{Ti} \min \left\{ R^\beta_{1,i}, R^\beta_{2,i}, \ldots, R^\beta_{N,i} \right\} \]

Sum over constituents of a jet

Distances to axes of N subjets

\( \tau_N \) measures departure from N-parton energy flow:

*if a jet has N subjets, \( \tau_{N-1} \) should be much larger than \( \tau_N \)
A jet with a small $\tau_{N,N-1}$ is more likely to have $N$ than $N-1$ subjets.

(from 1011.2268, with $\beta=1$)
Energy correlation functions

Probes of $N$-prong structures without requiring identification of subjets

$$ECF(N, \beta) = \sum_{i_1<i_2<\ldots<i_N\in J} \left( \prod_{a=1}^{N} pT_{i_a} \right) \left( \prod_{b=1}^{N-1} \prod_{c=b+1}^{N} R_{i_bi_c} \right)^{\beta}$$

Angular $(\gamma-\phi)$ distances between constituents

$ECF(N+1)$ is zero if there are only $N$ particles

More generally, if there are $N$ subjets one expects $ECF(N+1)$ to be much smaller than $ECF(N)$

[because radiation will be mainly soft/collinear to subjets]
Discriminators

\[ r^{(\beta)}_N \equiv \frac{ECF(N + 1, \beta)}{ECF(N, \beta)} \]

small for N prongs:
if N hard partons, small if radiation only soft-collinear

\[ C^{(\beta)}_N \equiv \frac{r^{(\beta)}_N}{r^{(\beta)}_{N-1}} = \frac{ECF(N + 1, \beta) ECF(N - 1, \beta)}{ECF(N, \beta)^2} \]

A jet with a small \( C_N \) is more likely to have N prongs and at most soft/coll radiation
Note different values of $\beta$
(chosen to maximise discriminating power)
A number of different IRC-safe jet algorithms exist
- They all try to be good proxies for hard partons, but they have different characteristics, especially with respect to soft particles

Jets from all algorithms inevitably suffer from pileup contamination
- Techniques exist to subtract it, either at jet-level, or at particle-level

Both the jet algorithms and many pileup subtraction techniques are packaged either in FastJet or in fjcontrib contributions
- Use of standard algorithms and packages (either directly or through interfaces) should be privileged, as it ensures reproducibility

The big news of the past few years has been the emergence of jet-based taggers and groomers, and more generally of jet substructure studies.

- They have proven their worth in ‘Standard Model’ analyses.
- They are being implemented in BSM searches.
- They are being used in heavy ions physics to probe the details of the parton splittings taking place in quenched jets.